REMARKS

By this Amendment, Applicant hereby adds claim 13. Accordingly, claims 5-13 are all of the claims pending in the application.

I. Formal Matter

The Examiner indicated that the Information Disclosure Statement submitted on August 1, 2008, was considered by the Examiner, however, the Examiner only returned the PTO/SB/08 form submitted with the Information Disclosure Statement filed on December 23, 2005.

Applicant respectfully requests that the Examiner initial and return the PTO/SB/08 form submitted with the Information Disclosure Statement filed on August 1, 2008.

II. Summary of the Office Action

Claims 5, 6, and 9 are newly rejected under 35 U.S.C. § 101. The Examiner maintained the rejection of claims 5-8 under 35 U.S.C. § 103(a) and also rejected newly added claims 9-12 under 35 U.S.C. § 103(a).

III. Claim Rejections under 35 U.S.C. § 101

Claims 5, 6, and 9 are rejected under 35 U.S.C. § 101 because the claimed invention is allegedly directed to non-statutory subject matter. The Examiner alleges that although claims 5, 6, and 9 recite a controller, the claims lack necessary physical articles or objects to constitute a machine or manufacture. The Examiner further alleges that the claims are non-statutory because they are functional descriptive material.

Applicant respectfully submits that claims 5, 6, and 9 are directed to statutory subject matter under 35 U.S.C. § 101. For example, claim 5 recites, *inter alia*, "a processor" and "a memory." Applicant respectfully requests that the Examiner withdraw the 35 U.S.C. § 101 rejection of claim 5 and dependent claims 6 and 9.

IV. Claim Rejections under 35 U.S.C. § 103(a)

Claims 5-12 are rejected under 35 U.S.C. § 103(a) as allegedly being unpatentable over U.S. Patent No. 5,349,518 to Zifferer et al. (hereinafter "Zifferer") in view of U.S. Patent Application Publication No. 2002/0147505 to Beck et al. (hereinafter "Beck") and further in view of U.S. Patent No. 5,886,274 to Jungleib (hereinafter "Jungleib"). Applicant respectfully traverses this rejection and respectfully requests the Examiner to reconsider this rejection at least in light of the comments which follow.

Turning first to independent claim 5, the Examiner alleges that Zifferer teaches "an instruction table for storing instructions and corresponding input/output types of parameters for the instructions," as recited, *inter alia*, in claim 5. Applicant respectfully disagrees.

The Examiner points to a database file 398 that, according to Zifferer, is where the software package maintains cross-reference information for the addresses 402. However, as argued in the Amendment filed on May 8, 2008, the database file 398 disclosed by Zifferer fails to correspond input/output types of parameters for instructions, with the instructions. Instead, according to Zifferer, each record 400 in the cross-reference file 398 is comprised of the PLC address 402, an identifier for each instruction that uses the address 404, a program file number containing the instruction 406, a rung number in the program file where the instruction is located 408, and other data 410 (*see* col. 7, lines 34-39 of Zifferer). Thus, Zifferer does not disclose that the input/output types of parameters for instructions are corresponded with the instructions in the cross-reference file 398. Instead, Zifferer appears to disclose that PLC addresses are corresponded with instructions that use the PLC addresses (including identifiers for each instruction and rung numbers where the instructions are located) (*see* col. 7, lines 34-39 of Zifferer); Zifferer makes no mention of input/output types with respect to the cross-reference file 398.

In response to Applicant's arguments for patentability, the Examiner responds by alleging that "Zifferer teach 'type' of file contain the same type of variable" (*see* page 12 of the Office Action). The Examiner points to col. 8, lines 25-40 of Zifferer. However, the Zifferer reference merely discloses that, when a user defines a symbol using auto-addressing, when a symbol 108 has been entered in the field 106 and the Enter key is pressed, a list of Data Table files 110 is displayed, wherein a Data Table file is highlighted (*see* col. 8, lines 21-29 of Zifferer). The file, according to Zifferer, is a Bit Data Table file as indicated by the "B" identifier 116 in the "Type" column (*see* col. 8, lines 29-32 of Zifferer).

Applicant respectfully notes that the "Type" column merely identifies the file type of a file, not the input/output types of parameters for instructions. Furthermore, the list of Data Table files 110 does not include instructions, thus the list of Data Table files is not the same as an instruction table that stores instructions and corresponding input/output types of parameters for the instructions.

While Zifferer discloses that some instructions in a ladder logic program 38 use only specific types of Data Table files, and with such instructions, the software package will list only valid Data Table files when a new symbol is being entered (*see* col. 9, lines 42-45 of Zifferer), Zifferer does not disclose that such Data Table files include an instruction table that stores both instructions and corresponding input/output types of parameters for the instructions. The Data Table files according to Zifferer, for example, do not store corresponding input/output types of parameters for instructions.

The Examiner further alleges that Zifferer teaches "a search/determination means for searching the instruction table for an instruction in a code in a portion of a sequence program selected as diversion data from an existing diversion-source sequence program, to determine a

corresponding input/output type of a parameter for the instruction," as recited, *inter alia*, in claim 5. Applicant respectfully disagrees.

The Examiner now points to the auto-addressing method of defining a symbol according to Zifferer as allegedly teaching this feature (*see* page 4 of the Office Action). According to Zifferer, when a symbol 108 has been entered in the field 106 and the Enter key is pressed, a list of Data Table files is displayed (*see* col. 8, lines 25-29 of Zifferer).

Applicant respectfully notes that Zifferer does not disclose determining a corresponding input/output type of a parameter for an instruction in a code in a portion of a sequence program selected as diversion data from an existing diversion-source sequence program. Instead of searching for an instruction in a code in a portion of a sequence program selected as diversion data, according to Zifferer, the auto-addressing is performed when a user is defining a new symbol (*see* col. 7 of Zifferer). Thus, rather than disclosing selected diversion data from an existing diversion-source sequence program, according to Zifferer, the user defines a new symbol.

Moreover, Zifferer fails to disclose determining an input/output type of a parameter based on a search of an instruction table. While Zifferer discloses that a list of Data Table files 110 is displayed, and the file types are indicated by identifiers 116 in the "Type" column (*see* col. 8, lines 25-32 of Zifferer), Zifferer does not disclose any correlation between the file type and the input/output type of parameters for instructions in the codes. In fact, Zifferer discloses that the instruction type may be different from the symbol type. For example, an instruction may be bit-oriented and the symbol-type may be word oriented. *See* col. 12, line 65 through col. 13, line 4 of Zifferer. Furthermore, according to Zifferer, a user can arbitrarily assign a file type to a Data Table file (*see* col. 9, lines 1-3 of Zifferer).

Additionally, searching an instruction table for an instruction in a code in a portion of a sequence program selected as diversion data from an existing diversion-source sequence program, to determine a corresponding input/output type of a parameter for the instruction, appears to be unnecessary for achieving the objectives according to Zifferer of attaching symbols to ladder logic instructions and element addresses (*see* col. 3, lines 46-68 of Zifferer).

The Examiner also alleges that Zifferer teaches "a search result creating/storing means for creating and storing a search result table by combining an address in the code in the selected portion of the sequence program, with the determined corresponding input/output type," as recited, *inter alia*, in claim 5. Applicant respectfully disagrees.

The Examiner points to disclosure in Zifferer at col. 7, line 50 through col. 8, line 40 related to defining a new symbol. The Examiner alleges that, based on an existing program, a "user creates database file contain symbols wherein maintain cross reference for the instructions and addresses and types" (*see* page 4 of the Office Action).

However, the database file 398 that maintains the cross-reference information disclosed by Zifferer fails to correspond input/output types of parameters for instructions, with the instructions. Instead, each record 400 in the cross-reference file 398 is comprised of the PLC address 402, an identifier for each instruction that uses the address 404, a program file number containing the instruction 406, a rung number in the program file where the instruction is located 408, and other data 410 (*see* col. 7, lines 34-39 of Zifferer). Thus, Zifferer does not disclose that the input/output types of parameters for instructions are corresponded with the instructions in the cross-reference file 398. Furthermore, as discussed above, the Data Table file according to Zifferer does not store corresponding input/output types of parameters for instructions. Thus, Zifferer does not disclose creating and storing a search result table.

The Examiner concedes that Zifferer does not teach "a component data creating means

for creating a variable data table by replacing the determined corresponding input/output types stored in the search result table with variable names," as recited, *inter alia*, in claim 5. However, the Examiner alleges that Beck discloses this feature. Applicant respectfully disagrees.

The Examiner now points to paragraph 0019 of Beck as allegedly disclosing this feature (*see* page 5 of the Office Action). However, Applicant respectfully notes that, according to Beck, symbolic input-output variables are replaced by the complete topological address of the corresponding input-output information (*see* paragraph 0013 of Beck), rather than by variable names.

Applicant respectfully notes that the Examiner has not pointed to any disclosure or teaching regarding "creating component data by adding the corresponding variable names to variables and to circuit information," as recited, *inter alia*, in claim 5. Applicant respectfully submits that this feature is not taught by any of the cited references.

The Examiner concedes that Zifferer and Beck do not teach "a component data diversion means for diverting the component data into an arbitrary position in a designated sequence program," as recited, *inter alia*, in claim 5. However, the Examiner alleges that Jungleib teaches this feature. Applicant respectfully disagrees.

As argued in the Amendment filed on May 8, 2008, the sequencer editor 335 according to Jungleib is limited to the function of modifications such as cut, paste, repeat, etc. of stored raw musical data 330 (*see* col. 4, lines 23-29 of Jungleib). Jungleib does not relate to a program sequence. Furthermore, Jungleib fails to teach or suggest a detailed description of diverting, such as what to divert, where to divert into, etc.

In response, the Examiner alleges that Jungleib teaches an original or existing sequence file (*see* page 13 of the Office Action). Applicant respectfully disagrees.

A person of ordinary skill in the art would clearly understand that a sequence program for a programmable logic controller is not the same as Musical Instrument Digital Interface (MIDI) technology and systems and methods for generating, distributing, storing, and performing musical work files (*see* col. 1, lines 7-10 of Jungleib). For example, sequence programs for programmable logic controllers are not musical compositions. Accordingly, Jungleib does not relate to sequence programs for programmable logical controllers or diverting component data into an arbitrary position in a designated sequence program.

Furthermore, as argued in the Amendment filed on May 8, 2008, Applicant respectfully submits that there is no apparent reason why a person of ordinary skill in the art at the time of the invention would have combined the Jungleib reference with Zifferer. Applicant further respectfully submits that the combination is improper because Jungleib is not analogous art. The Examiner indicates on page 5 of the Office Action that Jungleib teaches a sequence editor that preferably allows users to copy and paste a program sequence, and that Jungleib is analogous art because "they are from method of ladder logic programming for Programmable Logic Controller." Applicant respectfully disagrees.

Jungleib does not relate to a method of ladder logic programming for a PLC. Instead, according to Jungleib, Jungleib relates to a composition and playback system including a sound bank containing at least one instrument sound, an input device for receiving music control signals, a sequencer coupled to the input device for storing the music control signals, and a work manager coupled to the sound bank and to the sequencer for generating a musical work file containing the music control signals and at least a portion of the sound bank (*see* abstract of Jungleib). Accordingly, Applicant respectfully submits that there is no apparent reason why a person of ordinary skill in the art would combine Jungleib with Zifferer.

In response to Applicant's arguments, the Examiner alleges that Zifferer and Jungleib are analogous art because they are both related to sequence program editing tools (*see* page 14 of the Office Action). Applicant respectfully disagrees.

Zifferer relates to methods of developing ladder logic programs for Programmable Logic Controllers (PLCs) (*see* col. 1, lines 7-9 of Zifferer). Jungleib, on the other hand, relates to methods for generating, distributing, storing, and performing musical work files (*see* col. 1, lines 7-10 of Jungleib). As discussed above, a person of ordinary skill in the art would understand that sequence programs for programmable logic controllers are not musical works. Applicant further submits that programming is not the same as generating and editing musical compositions. At least for these reasons, Applicant respectfully submits that Jungleib is not analogous art.

At least for the above reasons, Applicant respectfully submits that claim 5 is patentable over Zifferer, Beck, and Jungleib. Independent claim 7 recites features similar to, although not necessarily coextensive with, the features discussed above with respect to claim 5. Accordingly, Applicant respectfully submits that claim 7 is patentable over Zifferer, Beck, and Jungleib at least for the reasons discussed above with respect to claim 5. Applicant respectfully submits that dependant claims 6, 9, and 11, and claims 8, 10, and 12 are patentable over Zifferer, Beck, and Jungleib at least by virtue of their dependency on claims 5 and 7, respectively.

Turning to claims 9 and 10, the Examiner alleges that Zifferer teaches "the input/output types of the parameters for the instructions comprise an input type, an output type, and an internal type," as recited, *inter alia*, in the claims. Applicant respectfully disagrees.

Zifferer discloses that all possible types of I/O modules are contained in the list 208 in FIG. 17. However, this list does not appear to include an internal type. Furthermore, this list shows module types, not parameter types for instructions.

At least for this reason, as well as for the reasons discussed above with respect to claim 5, Applicant respectfully submits that claims 9 and 10 are patentable over Zifferer, Beck, and Jungleib.

V. New Claim

Applicant hereby adds claim 13, which is supported throughout the specification. Applicant respectfully submits that claim 13 recites features similar to, although not necessarily coextensive with, the features discussed above with respect to claim 5. Accordingly, Applicant respectfully submits that claim 13 is patentable over Zifferer, Beck, and Jungleib at least for the reasons discussed above with respect to claim 5.

VI. Statement of Substance of Interview

Applicant thanks the Examiner for a courteous telephonic interview on November 25, 2008. The PTO-413 requires Applicant to file a Statement of Substance of Interview. The Statement of Substance of Interview is as follows:

The telephonic interview was conducted on November 25, 2008, with the following in attendance:

Cheng Yuan Tseng (Examiner) Eric S. Barr (Reg. No. 60,150)

Claims 5, 7, and 9 were discussed during the interview. The Examiner agreed that the claims are patentable over Zifferer, Beck, and Jungleib.

The Examiner alleged that claim 7 is not directed to statutory subject matter under 35 U.S.C. § 101 and suggested amending claim 7 to recite "creating and storing into component storage a search result table." Applicant hereby amends claim 7 to recite "creating and storing into a memory a search result table" (*see* FIG. 2 and page 8, lines 9-20 of the specification).

Applicant respectfully submits that claim 7 is tied to a machine and is directed to statutory subject matter under 35 U.S.C. § 101.

VII. Conclusion

In view of the above, reconsideration and allowance of this application are now believed to be in order, and such actions are hereby solicited. If any points remain in issue which the Examiner feels may be best resolved through a personal or telephone interview, the Examiner is kindly invited to contact the undersigned attorney at the telephone number listed below.

The USPTO is directed and authorized to charge all required fees, except for the Issue Fee and the Publication Fee, to Deposit Account No. 19-4880. Please also credit any overpayments to said Deposit Account.

Respectfully submitted,

/Eric S. Barr/

Eric S. Barr
Registration No. 60,150

SUGHRUE MION, PLLC
Telephone: (202) 293-7060
Facsimile: (202) 293-7860

WASHINGTON OFFICE
23373
CUSTOMER NUMBER

Date: December 1, 2008